

The Thesis committee for Itay Rosenzweig

Certifies that this is the approved version of the following thesis:

**Integrating hydrodynamic and oil spill trajectory
models for nowcasts/forecasts of Texas bays.**

**APPROVED BY
SUPERVISING COMMITTEE:**

Supervisor: _____

Ben R. Hodges

Paola Passalacqua

**Integrating hydrodynamic and oil spill trajectory
models for nowcasts/forecasts of Texas bays.**

by

Itay Rosenzweig, B.S.C.E.

Thesis

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Engineering

The University of Texas at Austin
August, 2011

Acknowledgements

I would like to begin with acknowledging several individuals and institutions without whom this work would not be possible. First my advisor, Professor Ben Hodges, who pointed me in the right direction and let me explore the subject on my own, but was always there to put me back on track when I wandered.

Dr. Buzz Martin and Dr. Dharhas Pothina respectively at the Texas General Land Office (which provided the funding for this research) and Texas Water Development Board were in touch with me during the entire two years of work to keep the project aimed toward the pressing real world goals they hope it will allow them to meet.

I would like to extend a very heartfelt thanks to Dr. Chris Barker and the team at the National Oceanographic and Atmospheric Administration Office of Response and Restoration who provided guidance on the use of their GNOME software. More importantly, they allowed me to visit their offices in Seattle during the summer of 2010 while they were constantly busy with the Deepwater Horizon spill, and still found the time to help me learn the GNOME software and talk to me about how this research would be applicable and invaluable to their daily work.

Finally, to the faculty and staff of the University of Texas at Austin Departments of Civil Engineering, and Environmental and Water Resources Engineering – thank you for guiding me during these years and for providing me with the resources necessary to complete this work.

**Integrating hydrodynamic and oil spill trajectory models for
nowcasts/forecasts of Texas bays.**

by

Itay Rosenzweig, MSE

The University of Texas at Austin, 2011

SUPERVISOR: Ben R. Hodges

A new method for automatically integrating the results of hydrodynamic models of currents in Texas bays with the National Oceanic and Atmospheric Administration's (NOAA) in house oil spill trajectory model, the General NOAA Operational Modeling Environment (GNOME), is presented. Oil spill trajectories are predicted by inputting wind and water current forces on an initial spill in a dedicated spill trajectory model. These currents can be field measured, but in most real and meaningful cases, the current field is too spatially complex to measure with any accuracy. Instead, current fields are simulated by hydrodynamic models, whose results must then be coupled with a dedicated spill trajectory model. The newly developed automated approach based on Python scripting eliminates the present labor-intensive practice of manually coupling

outputs and inputs of the separate models, which requires expert interpretation and modification of data formats and setup conditions for different models.

The integrated system is demonstrated by coupling GNOME independently with TXBLEND – a 2D depth-averaged model which is currently used by the Texas Water Development Board, and SELFE – a newer 3D hydrodynamic model with turbulent wind mixing. A hypothetical spill in Galveston Bay is simulated under different conditions using both models, and a brief qualitative comparison of the results is used to raise questions that may be addressed in future work using the automated coupling system to determine the minimum modeling requirements for an advanced oil spill nowcast/forecast platform in Texas bays.

Table of Contents

Acknowledgements.....	iii
Abstract.....	iv
Table of Contents.....	vi
List of Figures.....	viii
Chapter 1 – Introduction.....	1
1.1 Overview.....	1
1.2 Oil Spill Modeling.....	2
1.3 Models.....	4
1.4 Objectives.....	6
Chapter 2 – Methodology.....	8
2.1 Overview.....	8
2.2 Python and Object Oriented Programming.....	8
2.3 Approaches to Model Coupling.....	9
2.4 Python Wrapper.....	11
2.5 Galveston Bay.....	14
Chapter 3 – Results.....	18
3.1 Overview.....	18
3.2 Model Setup.....	19
3.3 Model Comparison and Discussion.....	21
Chapter 4 – Conclusions and Future Work.....	28
4.1 Conclusions.....	28
4.2 Future Work.....	29

Appendix A.....32

References.....38

List of Figures

Figure 1.2.1 – Workflow Process for Modeling an Oil Spill.....	3
Figure 2.4.1 – The Python Wrapper modules and structure.....	13
Figure 2.5.1 – Vertically out-of-scale representation of a Texas bay ship channel.....	14
Figure 2.5.2 – Image of Galveston Bay with ship channel.....	15
Figure 2.5.3 – 2D Depth Averaged Model of Corpus Christi Bay.....	16
Figure 2.5.4 – Flow in and over a ship channel using 3D SELFE.....	17
Figure 3.2.1 – Hourly wind record over Galveston Bay for Low and High Wind Conditions.....	20
Figure 3.3.1 – The effects of the different governing equations.....	21
Figure 3.3.2 – Testing 3D SELFE; the effects of model dimensionality.....	22
Figure 3.3.3 – The effects of the different wind forcing algorithms.....	23
Figure 3.3.4 – Testing 3D SELFE with low wind.....	24
Figure 3.3.5 – 3D SELFE modeled currents with low wind; peak flood and slack tide.....	25
Figure 3.3.6 – Testing 3D SELFE with high wind.....	26
Figure 3.3.7 – 3D SELFE modeled currents with high and low wind.....	27

Chapter 1 – Introduction

1.1 – Overview

This research presents a new approach to automated operation of oil spill and hydrodynamic models for Texas bays, part of an ongoing effort funded by the Texas General Land Office (GLO) in collaboration with the Texas Water Development Board (TWDB) to improve Texas' oil spill response. TWDB is responsible for operational modeling of the entire Texas coastline, working collaboratively with the National Oceanographic and Atmospheric Administration (NOAA) during an emergency. The existing operational model for bays uses the two dimensional (2D) TXBLEND hydrodynamic model to simulate the water currents, with manual data transfer and configuration of the General NOAA Operational Modeling Environment (GNOME) to forecast the oil spill trajectory. The present study develops an automated coupling system based on Python scripting to directly link the hydrodynamic and oil spill models. This coupled system is demonstrated by comparing GNOME trajectory simulations driven by currents from TXBLEND with simulations driven by currents from a newer three dimensional (3D) hydrodynamic model – SELFE. The coupled models are tested using Galveston Bay, the entrance to the Port of Houston, which is the second busiest port in the United States [Port of Houston Authority, 2011], and the site of spills of oil, fuel, and chemicals related to the oil industry every few months [Gold, 2009]. The Houston Ship Channel running through the otherwise shallow bay is a common bathymetric feature in Texas.

1.2 – Oil Spill Modeling

Oil spill modeling is fundamentally a three-stage process. The first stage is input data preparation, the second stage is current modeling (hydrodynamic modeling), and the third stage is particle tracking (spill trajectory modeling) [Wang et. al., 2005]. Input data preparation requires identifying all the operational parameters of the two models, including the model domains and grids, tidal boundary conditions, inflows, wind/rain and other weather data, and initial hydrodynamic conditions [Zhang & Baptista, 2008]. The hydrodynamic model is responsible for predicting the currents based on the forcings from the input data, while the spill trajectory model is responsible for applying these predicted currents as well as any other applicable forcings simulate the oil particles' fate and transport. In practice, the most common forcings applied by the spill trajectory model are currents, wind, diffusion and weathering/decay [Beegle-Krause, 2001]. The five steps of the workflow process for modeling an oil spill are displayed graphically in Figure 1.2.1. These steps include input data preparation, running the hydrodynamic model, converting current outputs to trajectory model inputs, running the trajectory model, and visualizing the spill simulation. Some applications exist in which some of the steps identified in Figure 1.2.1 are grouped together [Kerbaol & Collard, 2005]. One notable example is GNOME, a spill trajectory model which has its own built in visualization procedures [Beegle-Krause, 2001]. GNOME is the trajectory model used throughout this work.

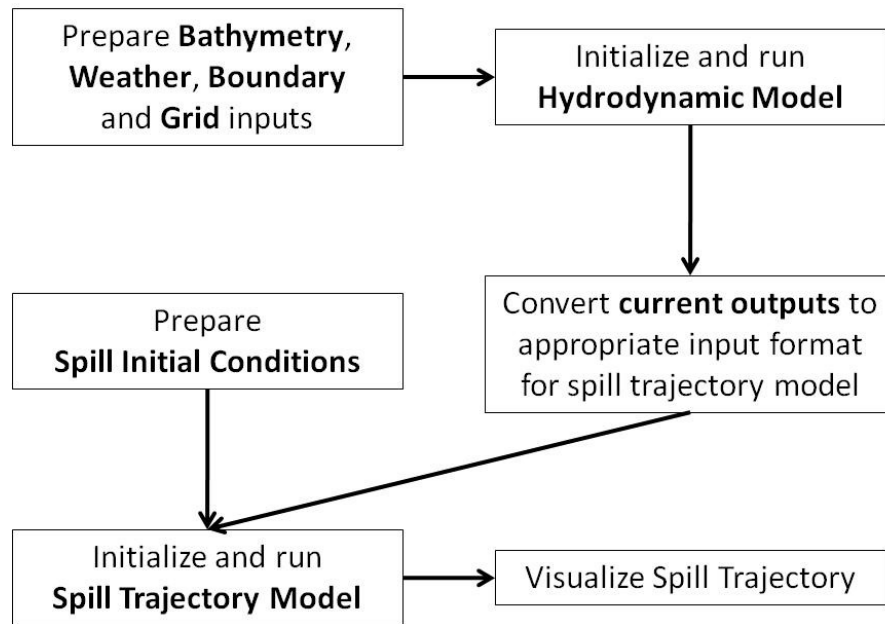


Figure 1.2.1 – Workflow Process for Modeling an Oil Spill

The TWDB is the Texas state government body responsible for monitoring and responding to oil spills in Texas bays [Crockett, 2010]. Presently, the operational hydrodynamic model is TXBLEND. For the TWDB operational system, wind data are obtained from the National Centers for Environmental Prediction Eta Model. This reanalysis model provides both gridded wind forecasts and reanalyzed hindcasts, the latter using interpolated field observations [Black, 1994]. Tide forecasts are based on tidal harmonic constituents, and hindcasts use field observations from the Texas Coastal Ocean Observation Network (TCOON) [Crockett, 2010].

1.3 – Models

GNOME

GNOME is NOAA's in-house oil spill trajectory model. Because NOAA is responsible for providing information to promote safe and effective response to all oil spills in US surface waters [NOAA OR&R, 2011] GNOME has been developed and tested in many different environments, from inland lakes to open ocean. GNOME has even been adopted by the international user community [Basar et. al., 2006], and is a de facto standard for operational oil spill modeling.

GNOME is a Lagrangian particle tracking model that transports simulated oil “particles” called Lagrangian Elements (LEs). Typical forces applied to transport these LEs are the current forces, direct wind forces, diffusion and decay [Beegle-Krause, 2001]. Unlike many transport problems, oil on the water surface does not act as a passive tracer moving at the current speed. Instead, it forms a surface layer over the water that interacts with both water and atmosphere, which may impede the development of wind-driven currents [Task committee on the modeling of oil spills, 1996]. However, for the smaller spatial scales and short durations associated with oil spills in a confined bay, this issue is arguably of limited concern.

The standard application of GNOME uses a graphical user interface (GUI) that is integral to the code and cannot be readily removed. GUIs are manual by nature, but this project developed a computer code (Python Wrapper) that automates the GUI input so the user can be removed from the data conversion by using the

alternative command file interface for GNOME. All GNOME commands available through the GUI can be given to GNOME through the command file. On running the GNOME executable, GNOME will operate all commands in the command file located in the same directory as the executable [Beegle-Krause, 2001].

TXBLEND

TXBLEND is the current operational model for TWDB. It is based on the BLEND model and was adapted for application in Texas waters by the TWDB Environmental Section [Crockett, 2010]. TXBLEND is a two dimensional (2D) depth averaged horizontally unstructured grid model which solves the continuity and momentum equations, and the advection-diffusion equations for conservation of salt. TXBLEND has an optional empirical wind forcing algorithm to drive the currents [Matsumoto, 1993].

SELF

SELF is a **S**emi-Implicit **E**ulerian **L**agrangian **F**inite **E**lement model. It is a three dimensional (3D) horizontally unstructured grid model with hybrid S-Z vertical coordinates [Zhang & Baptista, 2008]. Although initially developed specifically for the Columbia River estuary, the model has since been successfully applied to the Chesapeake Bay [Gong et. al., 2009], the Ria de Aveiro in Portugal [Rodrigues et. al., 2009] and the Guadiana Estuary [Oliveira et. al., 2006] among others.

SELFIE uses the Generic Length Scale turbulence closure which encompasses the 2.5-equation closure models (κ - ϵ , κ - ω , and Mellor-Yamada 2.5) [Zhang & Baptista, 2008]. This algorithm mixes turbulent wind energy through the free surface. Testing of this wind forcing algorithm against previously existing algorithms determined that the most significant effects are observed on the surface currents [Umlauf & Burchard, 2003], which are the currents most relevant to oil spill trajectory modeling.

SELFIE is a 3D model with hybrid S-Z (optional Z) coordinates. Although surface oil spill trajectory models like GNOME require only the 2D surface current field, accurate representation of the surface currents is only possible with accurate representation of the subsurface [Wang & Shen, 2010]. Studies of hypoxia in Corpus Christi Bay indicate that TXBLEND does not accurately predict behavior in the subsurface [Furnans, 2004]. A 3D model which allows subsurface layers to interact with, but behave separately from, the surface may more accurately represent subsurface, and therefore the surface.

1.4 – Objectives

The primary objective of this study is to develop a Python wrapper coupling the hydrodynamic models TXBLEND and SELFIE with GNOME. The wrapper is designed to minimize the manual steps required to run the models, and ensure any necessary manual steps are entirely pre-processing, that is, completed prior to

starting a model run. Use of the wrapper is demonstrated by a series of simulations of TXBLEND and SELFIE coupled with GNOME for a hypothetical spill in Galveston Bay. These simulations are used to raise questions about the modeling requirements for a future nowcast/forecast system for Texas bays.

Chapter 2 – Methodology

2.1 – Overview

The GNOME GUI (discussed in section 1.3) may be time consuming and susceptible to user input error, which are disadvantages for emergency response. Oil spill response procedures must be fast and based on reliable and accurate forecasts to prevent spills in Texas bays from reaching ecologically sensitive shorelines. Ensuring the availability of such forecasts requires someone whose primary job it is to be conversant in model inputs and outputs, data formats for multiple models, and the GNOME interface. This study develops an automated approach to coupling results of the hydrodynamic models TXBLEND and SELFE to GNOME. Python scripting was used to achieve a loosely coupled link between the models.

2.2 – Python and Object Oriented Programming

Python is an open source programming language that emphasizes modularity, readability, and speed. It is a “just-in-time” compiled language, which means code does not need to be compiled before it is run. It is compiled as part of the run process. Because of this, it is sometimes called an “uncompiled” language [Lutz, 2011]. For the purposes of this research, Python has two features that are invaluable. First, it can be used as a scripting language. It can, among other things, open files, create and write files, save files, and run executables. All of this is necessary to develop a Python wrapper program to automate the process of oil spill modeling.

Second, the modular approach lends itself very well to wrapping each step of the workflow originally shown in Figure 1.2.1, creating a series of modules which can then be combined to form a single master wrapper capable of running all of those steps automatically.

This treatment, wherein each module is an object, and the objects are allowed to interact with each other through a set of rules defined by the master wrapper, is a straight-forward application of object oriented programming [Mitchell, 2003]. Such a hierarchy has several major advantages over traditional programming. First, because each piece of code is an independent module, they can be switched out with ease, usually just by changing a single line in the master wrapper. When switching from TXBLEND to SELFE, there isn't a whole new program to be written, just a few modules to swap. Also, the rules that define how the objects interact with each other prevent bugs in one section of code from "leaking" into other sections, which can significantly reduce error checking and debugging times over traditional programming techniques [Lutz, 2011].

2.3 – Approaches to Model Coupling

The past few years have seen significant effort invested in the idea of integrating different types of models. Integrated model systems can be tightly coupled, coupled through an interface, or loosely coupled. In tightly coupled systems, the model integration is accomplished by rewriting one model to directly

incorporate the other as a submodel within the same code and data structure. Tight model coupling customizes the integration of the individual models, which usually translates to the most efficient computational implementation. The two submodels may exchange information without the inefficiencies of writing to a file, as is typically used in looser model coupling. The disadvantage of tight coupling is that it increases model development time and may make it difficult to debug and improve model components. Examples of tightly coupled model systems include large scale ocean circulation models with sea ice models [Mueller et. al., 2011], and coastal ocean models with particle tracking models [Fach & Klinck, 2006].

In interface coupling, the models to be coupled are developed to conform to a standard input/output format such as OpenMI [OpenMI Association, 2011]. The advantage of interface coupling is that it creates an environment in which many different models may exchange data through the standard interface. However, each model must either be originally developed to conform to the standard, or significant redevelopment work must be done to make legacy model codes compliant. Examples of coupling systems which use the OpenMI standard interface include an integrated agricultural, groundwater, and economic model [Bulatewicz et. al., 2010] and an integrated river basin management and hydrological model [Christensen, 2004].

Loosely coupled models exchange data through their native input and output formats. The output of one model is converted into the input of the next model. This approach is less computationally efficient than tight coupling, but is easier to develop

and debug. It is a less inclusive environment than interface coupling because the conversion from output to input formats must be accomplished individually for each model pair, but it significantly reduces development time when working with the many existing models which were never developed to be compliant with any standard interface. The Python Wrapper developed in this project automates the loose coupling between the hydrodynamic models and GNOME.

2.4 – Python Wrapper

The wrapper is equipped with modules for each of the steps of the spill modeling workflow except input preparation. All input data files, including the GNOME command file, must be prepared manually in advance of a model run. The modules are all operated by a master wrapper whose function is to call the routines contained in the operational modules in the correct order.

The first wrapper module checks that all required input data is available. If any expected input file is missing, the model run will not be allowed to begin, saving time by ensuring that no process begins that will terminate or crash before successful completion due to lack of data. The second module calls the hydrodynamic model executable. Output files are automatically saved to the same directory that contains the model executable. The third module converts the outputs from the hydrodynamic model to GNOME input and saves the new GNOME input file with the name expected by the user-prepared command file. Because each model has its own inputs,

its own executable, and its own output format, each model has a unique version of these three modules.

The fourth module is responsible for calling the GNOME executable from the directory with the command file. Because the GNOME input format is completely independent from the model origins of the information, this module is identical for all hydrodynamic models.

These modules automate the entire modeling process from the hydrodynamic modeling to the trajectory simulation. Spill visualization is accomplished using GNOME's native visualization capability. The only manual steps are the preparation of the input files for the hydrodynamic model, and the preparation of the GNOME command file. The structure of the Python wrapper is shown in Figure 2.4.1.

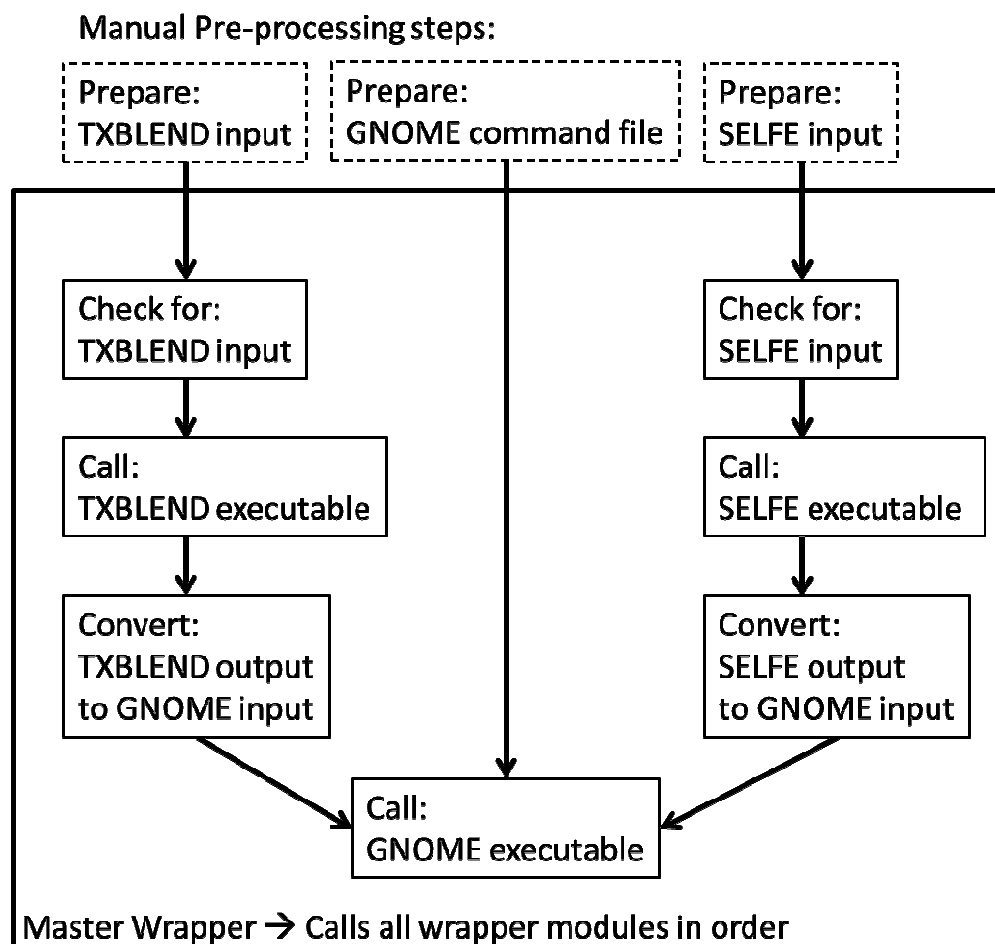


Figure 2.4.1 – The Python Wrapper modules and structure

2.5 – Galveston Bay

Most Texas bays, including Galveston and Corpus Christi, are fairly level and on the order of 10 ft. deep. The narrow ship channels dredged into the bottoms are on the order of 50 ft. deep and only 400 ft. wide, compared to the 15-30 mile width of the bays [Department of Commerce, 1998]. A vertically stretched representation of such a ship channel is shown in Figure 2.5.1, and a Google Satellite image of Galveston Bay with the ship channel identified is shown in Figure 2.5.2.

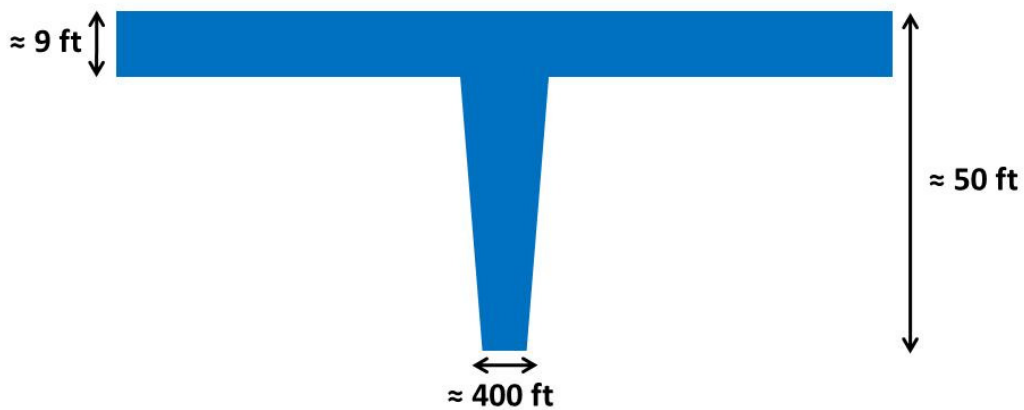


Figure 2.5.1 – Vertically out-of-scale representation of a Texas bay ship channel



Figure 2.5.2 – Image of Galveston Bay with ship channel [Google, 2011]

An accumulating body of evidence indicates that 2D models do not accurately capture the hydrodynamics of Texas bays [Kulis & Hodges, 2006 and Pothina, 2009]. A particular concern is the numerical artifacts introduced by the ship channels. Previous studies have shown that 2D models tend to produce a boundary in the surface currents along these channels [Furnans, 2004 and Pothina, 2009]. The depth-averaged velocity in the channel is dominated by the tidally driven along-channel flow in the depth of the channel, producing the boundary which can limit communication between currents on opposite sides of the ship channel [Pothina, 2009]. Figure 2.5.3 graphically demonstrates the decoupled circulations in Corpus

Christi Bay on opposite sides of the ship channel. The study which produced this figure was done to investigate hypoxia in the bay. While the information on hypoxia is irrelevant to the present study, the numerical effects on the circulation can clearly be seen running through the middle of the bay.

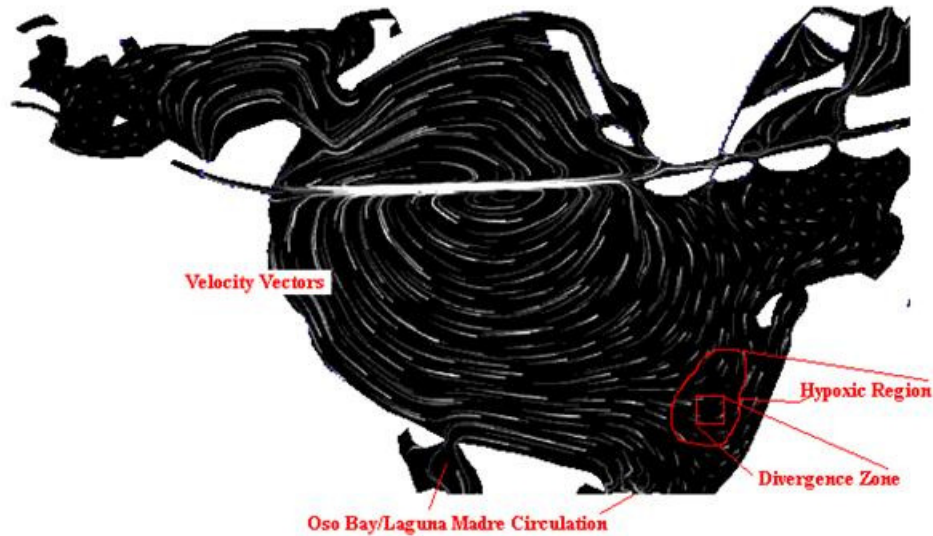


Figure 2.5.3 – 2D Depth Averaged Model of Corpus Christi Bay [Furnans, 2004]

Preliminary data from a field study at the Aransas Pass Tidal Inlet indicate that this boundary should not exist, and that currents above such a channel can have a significant cross channel component [Min, 2010]. Vertical grid resolution studies on a geometrically idealized ship channel in a shallow bay using SELFE show along channel currents deep in the channel, with cross channel currents on the surface above the channel [Pothina, 2009]. An example of such a result is shown in Figure 2.5.4.

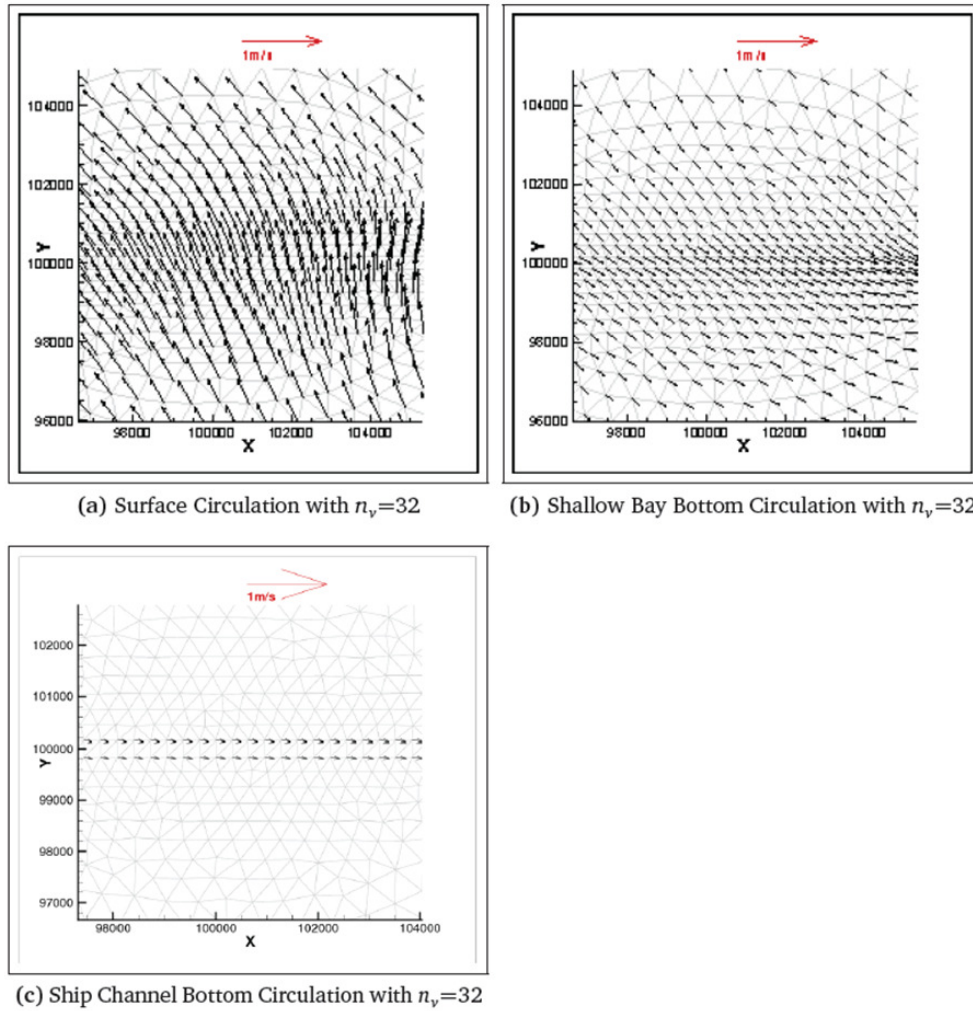


Figure 2.5.4 – Flow in and over a ship channel using 3D SELFЕ [Pothina, 2009]

Whether or not this numerical artifact significantly affects the accuracy of oil spill simulations may be determined by comparing the 2D TXBLEND model with the 3D SELFЕ model in the context of oil spill modeling.

Chapter 3 - Results

3.1 – Overview

Several recent studies developed and validated 3D hydrodynamic models against field data in different environments, showing them to be more accurate than simpler 2D models [Furnans, 2004 and Wang & Shen, 2010]. This project uses the Python wrapper described in Section 2.4 to compare a 2D model with a 3D model through the filter of oil spill modeling. Instead of just comparing the differences between the modeled current fields, using the automated link with GNOME allows for a very clear visualization of how the differences between the two hydrodynamic models manifest in different spill simulation results.

Three important differences between TXBLEND and SELFIE that are addressed in the tests below are difference in the governing equations and their discretization, different model dimensionality (2D vs. 3D), and different algorithms to drive wind forcing on currents. TXBLEND, 2D SELFIE, and 3D SELFIE were tested and compared under several wind conditions to determine the effects of differences in the models on spill simulations.

The tests are numbered as follows in the presentation of the results:

- 1 – 2D TXBLEND and 2D SELFE, No Wind
- 2 – 3D SELFE, No Wind (Compared with results of Test 1)
- 3 – 2D TXBLEND and 2D SELFE, Low Wind
- 4 – 3D SELFE, Low Wind
- 5 – 3D SELFE, High Wind

3.2 – Model Setup

To test the effects of the difference between the governing equations of the two models, 2D TXBLEND was run alongside 2D SELFE with no wind forcing. To test the effects of dimensionality, SELFE was run in 3D mode with no wind and compared to the runs of 2D TXBLEND and 2D SELFE with no wind. 3D SELFE was run with 10 S-layers.

To test the effects of the difference in wind forcing algorithms, TXBLEND and 2D SELFE were run for 48 hour simulations under Eta hindcast wind conditions for April 6-7, 2010. The winds during the two day period April 6-7 are defined as the low wind condition, where the average wind speed is 3.8 knots [TAMU 2010]. To test the sensitivity of the results to direct wind transport, 3D SELFE was also run during a higher wind event on May 1-2, 2010. The winds during the two day period May 1-2 are defined as the high wind condition, where the average wind speed is 10.2 kts [TAMU, 2010]. So that the different tidal series on each of these two day

periods would not confound the comparison, an artificial diurnal sinusoidal tide with amplitude of 1 ft. was input.

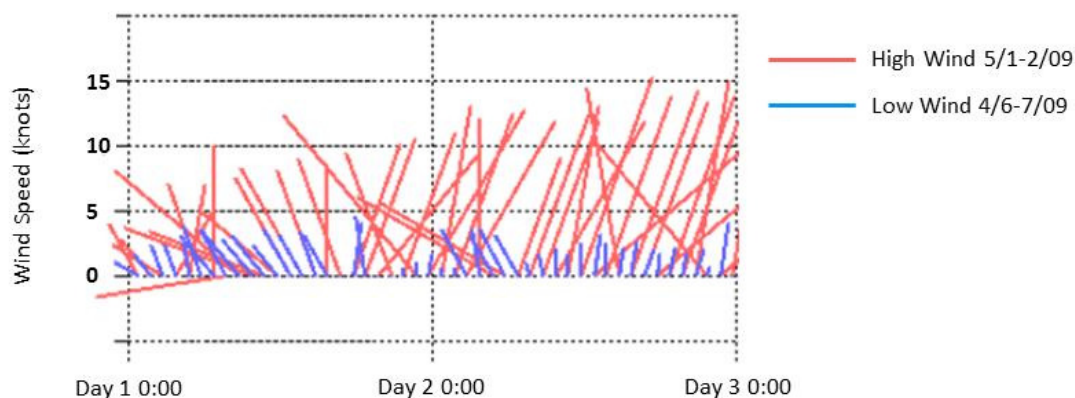


Figure 3.2.1 – Hourly wind record over Galveston Bay for Low and High Wind Conditions, [TAMU, 2010]

Finally, to compare the overall performance of the models against each other, 3D SELFE was run using the same wind condition as in Figure 4.1.1 and compared to the 2D TXBLEND and 2D SELFE runs.

Each test simulates an identical 100 barrel point source spill in the middle of the Galveston Bay ship channel. GNOME's built in visualization capabilities were used to create maps of the spill fate at the end of each 48 hour simulation. These maps were then overlaid on Google satellite imagery of Galveston Bay [Google, 2011].

3.3 – Model Comparison and Discussion

TEST 1: 2D TXBLEND and 2D SELFE, No Wind

Figure 3.3.1 shows the difference in simulated spill fate between GNOME runs driven by currents from 2D TXBLEND and 2D SELFE with no wind forcing. Since both models use the same horizontal grid, have only one vertical layer, and there is no wind forcing, the only difference between them is the difference in the governing equations and solution method. Although some small differences in spill fate are visible, they are likely negligible considering both numerical error and the error introduced by measured wind, tide, and bathymetry inputs.

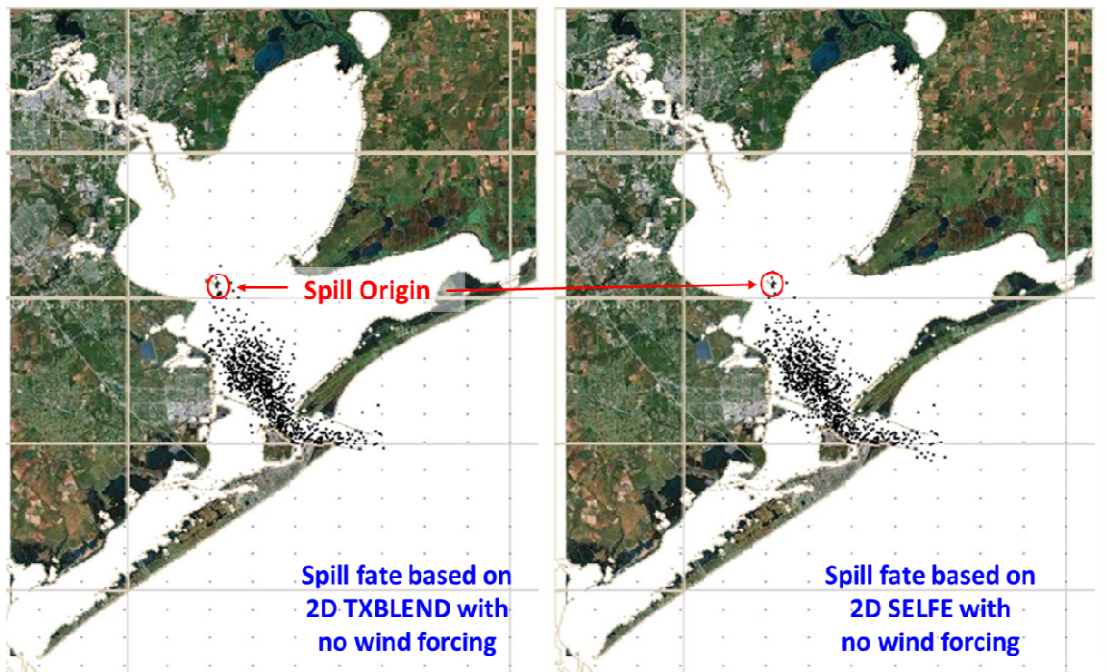


Figure 3.3.1 – The effects of the different governing equations

TEST 2: 3D SELFE, No Wind

Figure 3.3.2 shows the results of coupling GNOME with output from a 3D SELFE run with no wind. The difference between spill fates from this run and the two runs in Test 1 is visually obvious. If 3D models do in fact more accurately represent the real currents in Texas bays, any future development of a new oil spill nowcast/forecast platform should address the issue of model dimensionality.

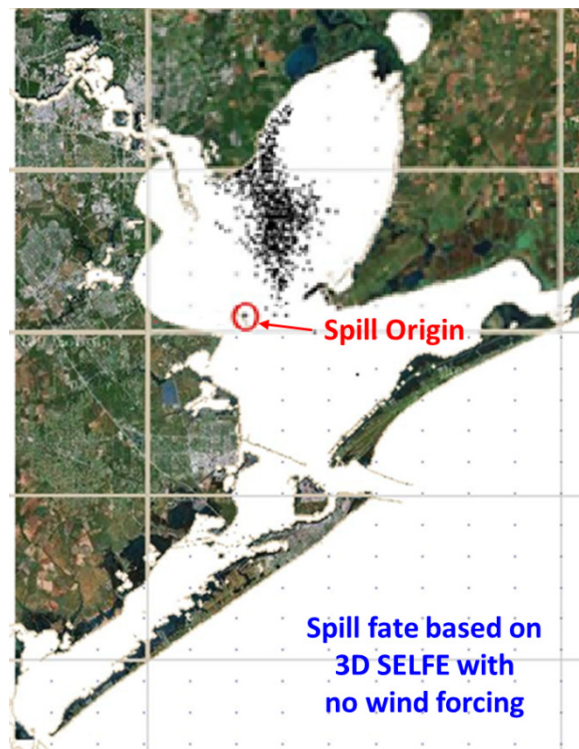


Figure 3.3.2 – Testing 3D SELFE; the effects of model dimensionality

TEST 3: 2D TXBLEND and 2D SELFE, Low Wind

Figure 3.3.3 shows the results of a TXBLEND run and a 2D SELFE run under the low wind condition defined in Figure 3.2.1. The difference between the two runs in this test is larger than Test 1, indicating that the difference in wind forcing algorithms is driving a change in the simulated spill fate. However, unlike the case of model dimensionality where evidence exists that 3D models are more accurate in Texas bays than 2D models (see Section 2.5), no studies have explicitly shown that SELFE's Generic Length Scale turbulence wind model is more accurate in simulating wind driven currents Texas bays than TXBLEND's empirical wind mixing algorithm.

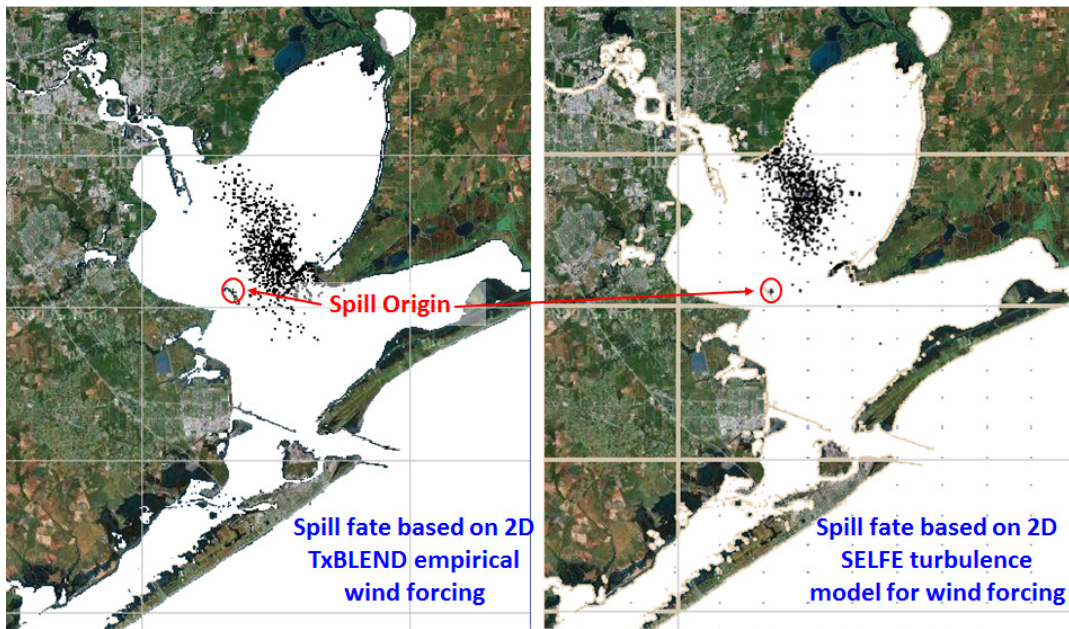


Figure 3.3.3 – The effects of the different wind forcing algorithms

TEST 4: 3D SELFE, Low Wind

Figure 3.3.4 shows the results of a 3D SELFE run under the low wind condition defined in Figure 3.2.1. This test reveals the combined effects of the differences in wind forcing algorithm and grid dimensionality on spill simulation results. When compared with results from Tests 2, this figure reveals that, when wind is present, there is little substantial difference between modeled spill fate based on currents from 2D SELFE and 3D SELFE.

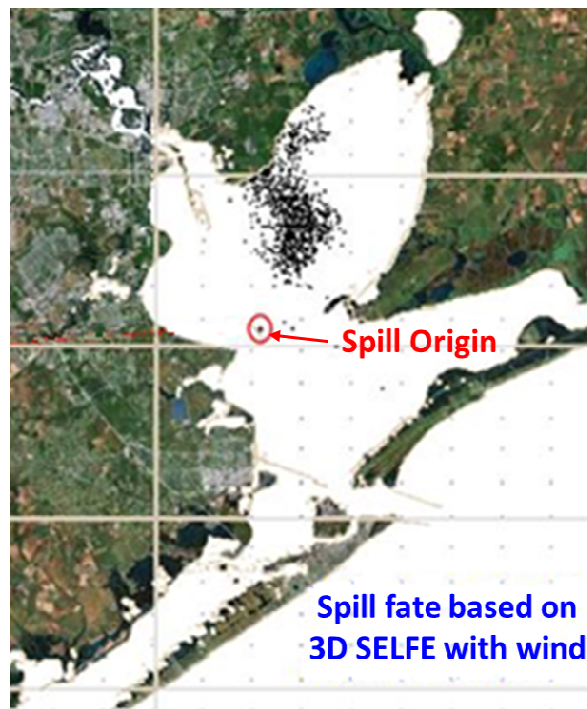


Figure 3.3.4 – Testing 3D SELFE with low wind

This result is different from the comparison between 2D SELFE and 3D SELFE in the absence of wind. One possible explanation for this was previously mentioned in

Section 1.3; namely, that the oil on the surface is not a passive tracer of the currents. GNOME transports the spill LEs by applying forces from the currents, and by applying direct forces from the wind. These forces are called “movers”. The default GNOME parameter is to transport LEs in the direction of the wind at 3% of the wind speed. At an average wind speed over the two day simulation of 3.8 kts, the average speed imparted to the oil directly from the wind by GNOME is 0.11 kts. Figure 3.3.5 shows that, even during the peak flood tide, the direct wind mover is the same order of magnitude as the currents in the body of the bay east of the channel. Near slack tide, the direct wind mover can be as large as the currents over the channel.

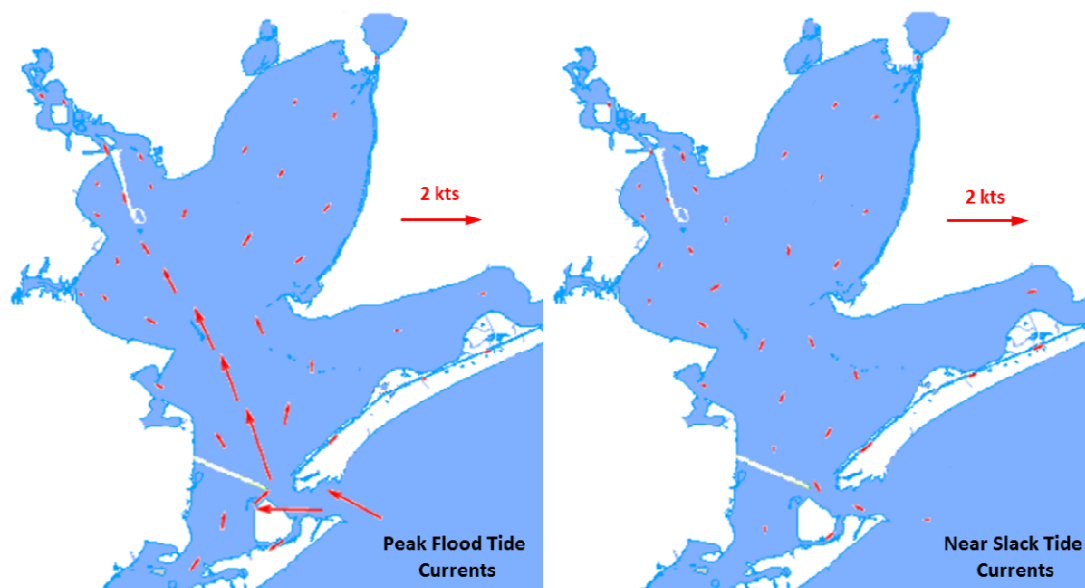


Figure 3.3.5 – 3D SELFE modeled currents with low wind; peak flood and slack tide

TEST 5: 3D SELFE, High Wind

Since the direct wind mover scales linearly with wind speed, it can become the dominant transporting force in higher wind scenarios. Figure 3.3.6 shows the results of a 3D SELFE run under the high wind condition defined in Figure 3.2.1.

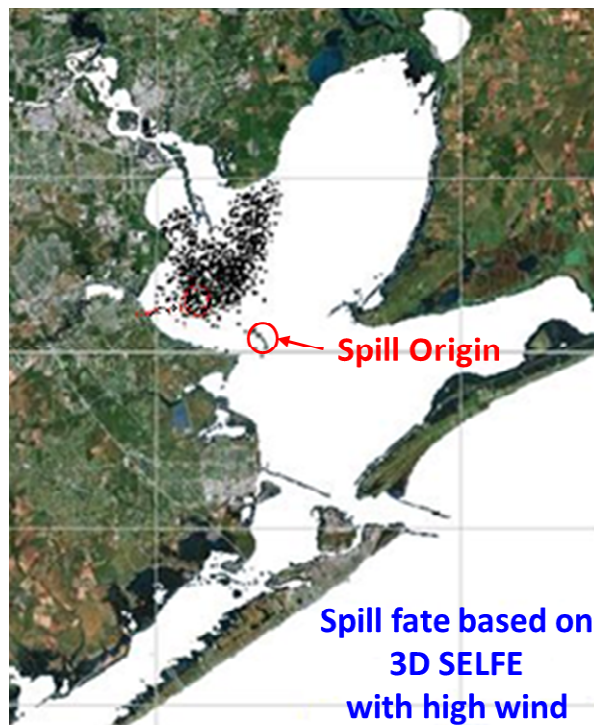


Figure 3.3.6 – Testing 3D SELFE with high wind

Figure 3.3.7 shows 3D SELFE modeled currents during slack tide for both the low and high wind conditions. Although there is a difference in the currents in between the high and low wind conditions, shown in Figure 3.3.7, the difference in spill

simulation results shown between Tests 4 and 5 is much larger than the small differences in currents would suggest. indicates that in the high wind condition, direct wind transport is in fact the dominant mechanism for spill transport.

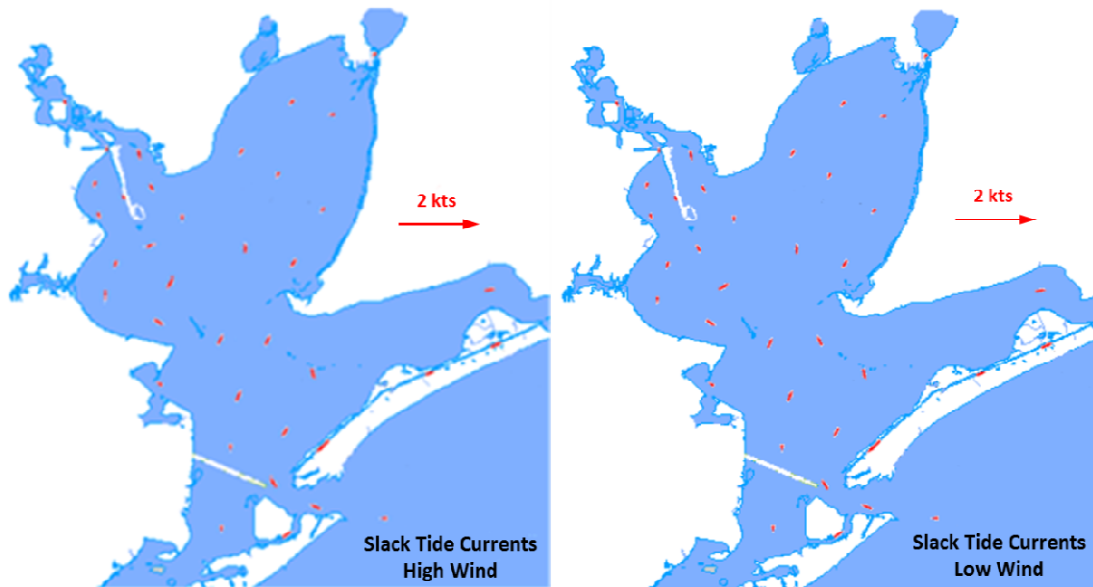


Figure 3.3.7 – 3D SELFE modeled currents with high and low wind

Chapter 4 – Conclusions and Future Work

4.1 – Conclusions

The Python wrapper developed for this project successfully automates the link between TXBLEND and SELFE with GNOME. Hydrodynamic model input and a GNOME command file must be manually prepared, but this preparation is in the nature of preprocessing, and once a model run is started, it is not interrupted for manual steps. Future work required to expand this wrapper into a fully automated system for nowcasts/forecasts in Texas bays is addressed in Chapter 6.

The four tests described in this study show that there is a difference between the spill simulation results when GNOME is coupled with TXBLEND and SELFE. 2D TXBLEND does not produce significantly different results from 2D SELFE in zero wind conditions. When wind is introduced, there is a difference between the simulated spill fate driven by TXBLEND and 2D SELFE, indicating that the different wind mixing algorithms need to be accounted for in future tests to determine which model is the best platform for oil spill modeling in Texas bays. The results from 3D SELFE were very different from either of the 2D models in the absence of wind, indicating that the surface current mover is highly dependent on model dimensionality. However, this appears not to be the case once wind is introduced into the models. Because a surface oil spill does not act as a passive tracer of the currents, wind driven transport may be the dominant transport mechanism for surface oil spills, especially in high wind conditions.

4.2 – Future Work

To continue the development of a fully automated system for nowcasts/forecasts in Texas bays, the next stage is automated data collection. The wrapper in its present state still requires users to manually create input files for the models. The TCOON network is already capable of streaming live tidal and inflow data, and live wind data and forecasts are publicly available through services such as NOAA's National Data Buoy Center [US Department of Commerce, 2011]. Using these live data sets, the wrapper can be expanded to collect the live data and create the necessary input files automatically. A version of this idea is already running at the TWDB, but only with tidal and inflow data, and using TXBLEND. TXBLEND runs constantly in the background, producing continually updated nowcasts. If this were extended to producing continuously updated forecasts, oil spill responders would only need to input initial spill conditions (location, time, size), and the fast Lagrangian trajectory model would collect forecast current outputs from the hydrodynamic model via the Python wrapper and provide a near-instant trajectory forecast. The final puzzle piece in the development of this platform would be to host it on a thin client – making it accessible from any internet browser without downloading any data. Spill responders in the field could hear about a spill, log on to the server which runs the model, input the latest data they have on spill initial

conditions, and be ready to get on the boats and respond before the ship that spilled the oil even reaches shore.

The tests of the Python wrapper conducted in this study were used to do a very preliminary comparison of the TXBLEND and SELFIE models in the context of oil spill modeling. However, the true magnitude and importance of the differences between the models can only be properly revealed with a more robust statistical comparison between the models. Future tests to compare the models should answer:

- In what conditions does wind become the dominant transport mechanism?
- Which model is least sensitive to errors in the measured inputs?
- Are the results of the tests in this study unique to Galveston Bay?

Or do they apply generally to shallow bays with ship channels?

- If it is determined that a 3D model is required –

Are spill simulation results sensitive to vertical resolution?

Answering these questions will require a large number of hydrodynamic model – GNOME linked runs. The automated coupling method embodied in the Python wrapper can be a very powerful tool in reducing the amount of time required to perform all of these runs. All of the manual work of preparing the different inputs and arranging them in the proper directories would be done up front. A simple script would instruct the Python wrapper to run in each of the directories. Once the inputs

are prepared and the wrapper is started, there would be no interruptions until all output spill simulations are ready for comparison.

Appendix A

This appendix provides a closer look at the code and structure of the Python wrapper. The code is reproduced with comments explaining the function of each section. The code shown is for a SELFE run. Note that exactly corresponding modules exist for TXBLEND runs.

MODULE 0: Master Wrapper

```
# master_wrapper.py
# This is the master wrapper which calls all the other
# modules.

# This first section imports all of the functional
# modules. It allows the master wrapper to call all
# of the functions within the modules

import selfe_in          # or txblend_in
import run_selfe         # or run_txblend
import selfe_to_gnome    # or txblend_to gnome
import run_gnome

# This second section runs the functions defined in the
# imported modules.

selfe_in.check()
run_selfe.run()
selfe_to_gnome.main()
run_gnome.run()
```

MODULE 1: Checking Input Files

```
# selfe_in.py
# This module checks for all SELFЕ input files.

def check() # Creates a function that the master
            # wrapper can call on

    file1 = open('hgrid.gr3', 'r')
    file2 = open('param.in', 'r')
    file3 = open('vgrid.in', 'r')
    file4 = open('interpol.gr3', 'r')

    # These are the minimum required
    # SELFЕ input files. SELFЕ will not
    # run without these files. The
    # module does not check for optional
    # inputs.

    file1.close()
    ...

    # List all the files with the
    # '.close()' argument.

# The logic behind this segment is that the function
# cannot open a non-existent file in 'read only' mode.
# If the necessary file does not exist and the wrapper
# fails to open it, the entire program will stop and
# display an error.
```

MODULE 2: Running Hydrodynamic Model

```
# run_selfe.py
# This module runs the SELFЕ executable.

# The Python library module 'subprocess' is used to
# write commands that mimic the terminal interface.

def run()

    import subprocess
    subprocess.Popen('./selfe.exe')

    # This format will only work if the
    # model executable is located in the
    # working directory.
```

MODULE 3: Converting Hydrodynamic Model Output to GNOME Input

```
# selfe_to_gnome.py
# This module converts SELFЕ current outputs to GNOME
# inputs.

def main()

    filelevel = open('*_hvel.64', 'r')
                # SELFЕ horizontal velocity outputs

    filemap = open('hgrid.gr3', 'r')
    fileparam = open('param.in', 'r')

    filein = open(__GNOME current file__, 'w')
                # User named GNOME input file

    filein.write('[FILETYPE] "PTCUR"')
                # GNOME input file header

    for i in range(6)
        filelevel.readline()

    filelevel.readline() = nvrt

    for i in range(nvrt+1)
        filelevel.readline()

    filemap.readline.format(np, ne)
    str1 = str('Vertices', np, 0)
    filein.write(str)

    for i in range(np)

        filemap.readline.format(pt, x, y, extra)
                # Reading the list of nodes from
                # SELFЕ input

        str2 = str(pt, x, y)
        filein.write(str2)
                # Rewriting in GNOME map format
```

```

for i in range(np+ne+2)
    filelevel.readline()

for line in fileparam
    # Using parameter input to calculate
    # output time steps

    if 'RNDAY' in line
        file.readline.format(rnday, extra)
    if 'Dt' in line
        file.readline.format(dt, extra)
    if 'output every' in line
        file.readline.format(outstep, extra)

ntimeout = rnday*3600*24 / (dt*outstep)

for i in range(ntimeout)
    # GNOME Time Data Block for each
    # output time step

    filelevel.readline() = time
    str3 = str(time)
    filein.write('[TIME]', time)

    for j in range(np)
        filelevel.readline.format(ptnum, u, v, w)
        # Reading SELFIE 3D vector output

        str4 = str(ptnum, u, v)
        filein.write(str4)
        # Rewriting as GNOME 2D vector on
        # surface layer

```


MODULE 4: Running GNOME

```
# run_gnome.py
# This module runs the GNOME executable
# and saves an screen capture of the final spill map

def run()

    import subprocess

    subprocess.Popen('./gnome.exe')

# The './' command ensures that the copy of the GNOME
# executable in the working directory is called.
# This is necessary to ensure that GNOME identifies the
# prepared command file.
```

References

- Başar, E., Köse, E., and Güneroglu, A. (2006). "Finding risky areas for oil spillage after tanker accidents at Istanbul strait." *International Journal of Environment and Pollution* 27(4): 388–400.
- Beegle-Krause, C.J. (2001). "General NOAA Oil Modeling Environment (GNOME): A New Spill Trajectory Model." IOSC 2001 Proceedings Vol. 2: 865-871.
- Black, T. L. (1994). "The new NMC mesoscale Eta Model: Description and forecast examples." *Weather Forecasting* 9: 265-278.
- Bulatewicz, T., Yang, X., Peterson, J.M., Staggenborg, S., Welch, S.M., and Steward, D.R. (2010). "Accessible integration of agriculture, groundwater, and economic models using the Open Modeling Interface (OpenMI): methodology and initial results." *Hydrology and Earth System Sciences* 14: 521-534.
- Christensen, F.D. (2004). "Coupling Between the River Basin Management Model (MIKE BASIN) and the 3D Hydrological Model (MIKE SHE) with the use of the OpenMI system." Proceedings of the 6th International Conference on Hydroinformatics.
- Crockett, D. (2010). "The Texas Water Development Board: Hydrodynamic and Oil Spill Modeling" Texas Water Development Board Website: <http://midgewater.twdb.state.tx.us/bays_estuaries/bhydpge.html>.
- Department of Commerce (1998). "Texas Coast (G200) Bathymetric Digital Elevation Model (30 Meter Resolution)." NOAA Ocean Service Special Projects Online Archive: <<http://estuarinebathymetry.noaa.gov>>.
- Fach, B.A., and Klinck, J.M. (2006). Transport of Antarctic krill (*Euphausia superba*) across the Scotia Sea. Part I: Circulation and particle tracking simulations." *Deep Sea Research Part I: Oceanographic Research Papers* 53(6): 987-1010.

Furnans, J. (2004). "CRWR Online Report 04-3: Exploring Hydrodynamic Modeling of Texas Bays with focus on Corpus Christi Bay & Lavaca Bay." CRWR Online Report: <<http://www.cwr.utexas.edu/online.shtml>>.

Gong, W., Shen, J., Cho, K.H. and Wang, H.V. (2009). "A numerical model study of barotropic subtidal water exchange between estuary and subestuaries (tributaries) in the Chesapeake Bay during northeaster events." *Ocean Modelling* 26(3-4): 170-189.

Google (2011). Google Maps Website: <<http://maps.google.com>>.

Kerbaol, V. and F. Collard (2005). "SAR-derived coastal and marine applications: From research to operational products." *Ieee Journal of Oceanic Engineering* 30(3): 472-486.

Kulis, P.S. and B. R. Hodges (2006), "Modeling gravity currents in shallow bays using a sigma coordinate model." *Proceedings of the 7th International Conference on Hydrosience and Engineering, Philadelphia, PA, Electronic Proceedings*: <<http://hdl.handle.net/1860/1458>>.

Lutz, M. (2011). Programming Python, Fourth Edition O'Reilly Media. ISBN-13: 978-0596158101.

Matsumoto, J. (1993). "User's Manual for the Texas Water Development Board's Hydrodynamic and Salinity Model, TxBLEND." Texas Water Development Board, Austin, Texas.

Min, D. (2010). "Characteristics of ocean currents and water properties observed throughout the water column at the Aransas Pass Tidal Inlet during Summer/Fall 2010." Gulf Estuarine Research Society Meeting Fall 2010.

Mitchell, J.C. (2003). Concepts in programming languages Cambridge University Press. ISBN 0-521-78098-5

Mueller, J., Wagner, A., Fahl, K. et. al. (2011). "Towards quantitative sea ice reconstructions in the northern North Atlantic: A combined biomarker and numerical modelling approach." *Earth and Planetary Science Letters* 306(3-4): 137-148.

US Department of Commerce (2011). “National Data Buoy Center.” NDBC Website: <<http://www.ndbc.noaa.gov/>>.

NOAA OR&R (2011) “Office of Response and Restoration: Responding to Oil Spills.” NOAA Office of Response and Restoration Website: <<http://response.restoration.noaa.gov/>>.

Oliveira, A., Fortunato, A.B. and Pinto, L. (2006). “Modeling the Hydrodynamics and the Fate of Passive and Active Tracers in the Guadiana Estuary.” *Estuarine, Coastal and Shelf Science* 70: 76-84.

OpenMI Association (2011). “What is OpenMI?” OpenMI Association Website. <<http://www.openmi.org>>.

Port of Houston Authority (2011). “General Information: The Port of Houston.” Port of Houston Authority Website: < <http://www.portofhouston.com/>>.

Pothina, D. (2009) “A multimodel approach to modeling bay circulation in shallow bay - ship channel systems (Doctoral Dissertation).”

Rodi, W. (1984). “Turbulence models and their applications in hydraulics: a state of the art review.” International Association for Hydraulics Research, Delft, The Netherlands.

Rodrigues, M., Oliveira, A., Queiroga, H., Fortunato, A.B. and Zhang, Y.J. (2009). “Three-Dimensional Modeling of the Lower Trophic Levels in the Ria de Aveiro (Portugal).” *Ecological Modelling* 220: 1274-1290.

TAMU Department of Atmospheric Sciences (2010). “Weather Interface.” TAMU Dept. of Atm. Sci. Website: < http://leonardo.met.tamu.edu/Weather_Interface/>. Last Accessed: MAY 3 2010.

Task committee on modeling of oil spills of the water resources engineering division. (1996). “State-of-the-art review of modeling transport and fate of oil spills.” *Journal of Hydraulic Engineering-ASCE* 122(11): 594-609.

Umlauf, L., Burchard, H. (2003). "A generic length-scale equation for geophysical turbulence models." *Journal of Marine Research* 6: 235–265.

Wang, J.H. and Shen, Y.M. (2010). "Development of an integrated model system to simulate transport and fate of oil spills in seas." *Science China – Technological Sciences* 53(9): 2423-2434.

Wang, S. D., Y. M. Shen and Y. H. Zheng (2005). "Two-dimensional numerical simulation for transport and fate of oil spills in seas." *Ocean Engineering* 32(13): 1556-1571.

Zhang, Y.-L. and Baptista, A.M. (2008). "SELF: A semi-implicit Eulerian-Lagrangian finite-element model for cross-scale ocean circulation", *Ocean Modeling* 21(3-4): 71-96.